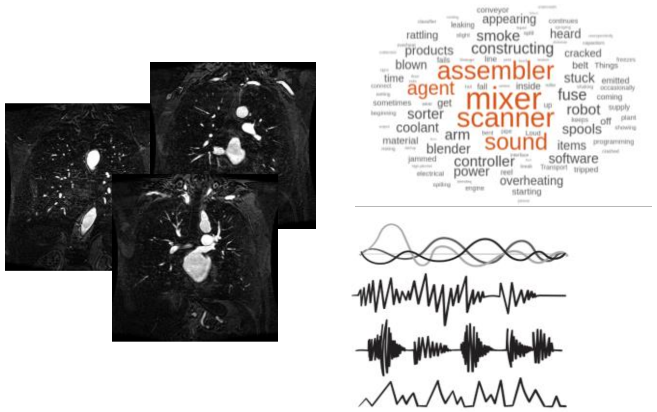
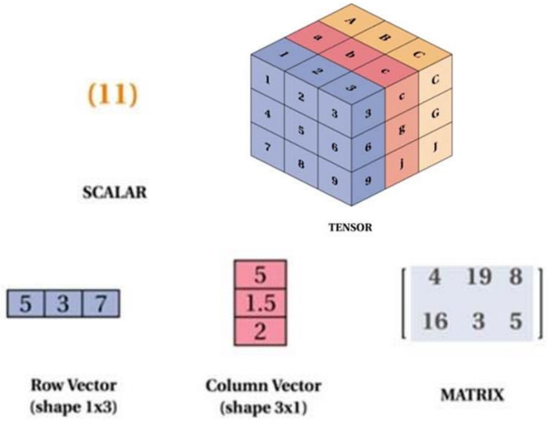
Introduction to Deep learning

- Deep Learning is a subset of machine learning
    - It uses multi-layered neural networks to model and understand **complex patterns** in data.



Raw data

→



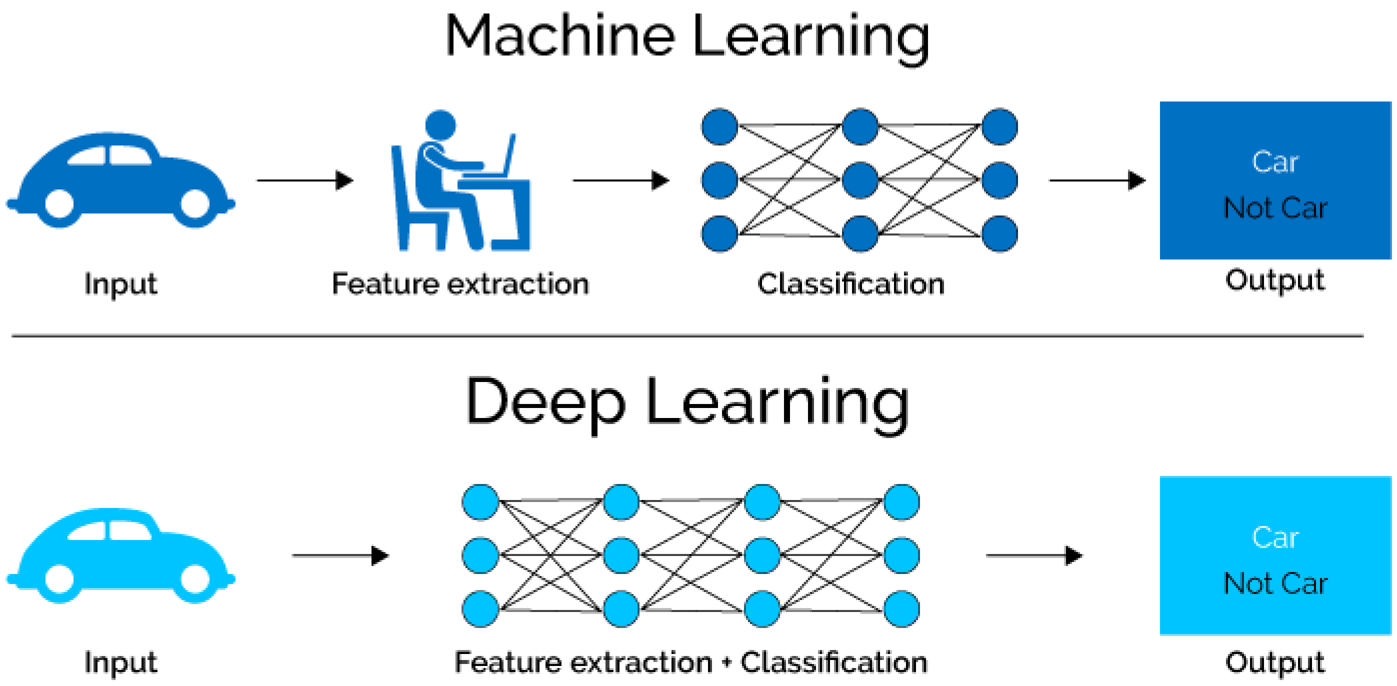
Representation

→

Learning Task

Deep learning vs Traditional Machine learning

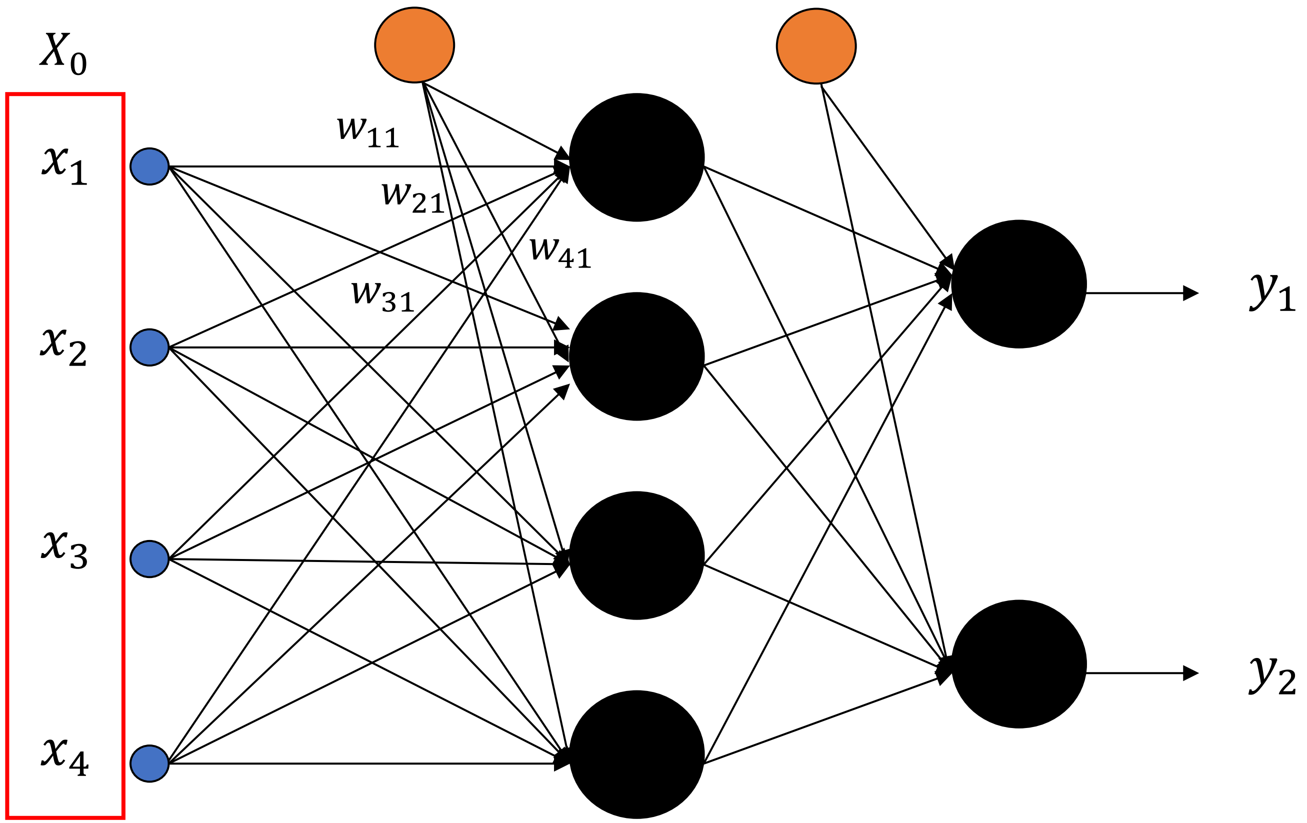
- Feature engineering had a key role in **ML**
  - Hand-crafted features (e.g., word co-occurrence, term frequency)



Data driven features

- **Deep neural network (DNN) as feature/representation learner**

A Simple Neural Network (NN)



$$\boxed{W_1 = \begin{bmatrix} w_{11} & \cdots & w_{41} \\ \vdots & \ddots & \vdots \\ w_{14} & \cdots & w_{44} \end{bmatrix} \qquad W_2 \rightarrow (4,2)}$$

W_i

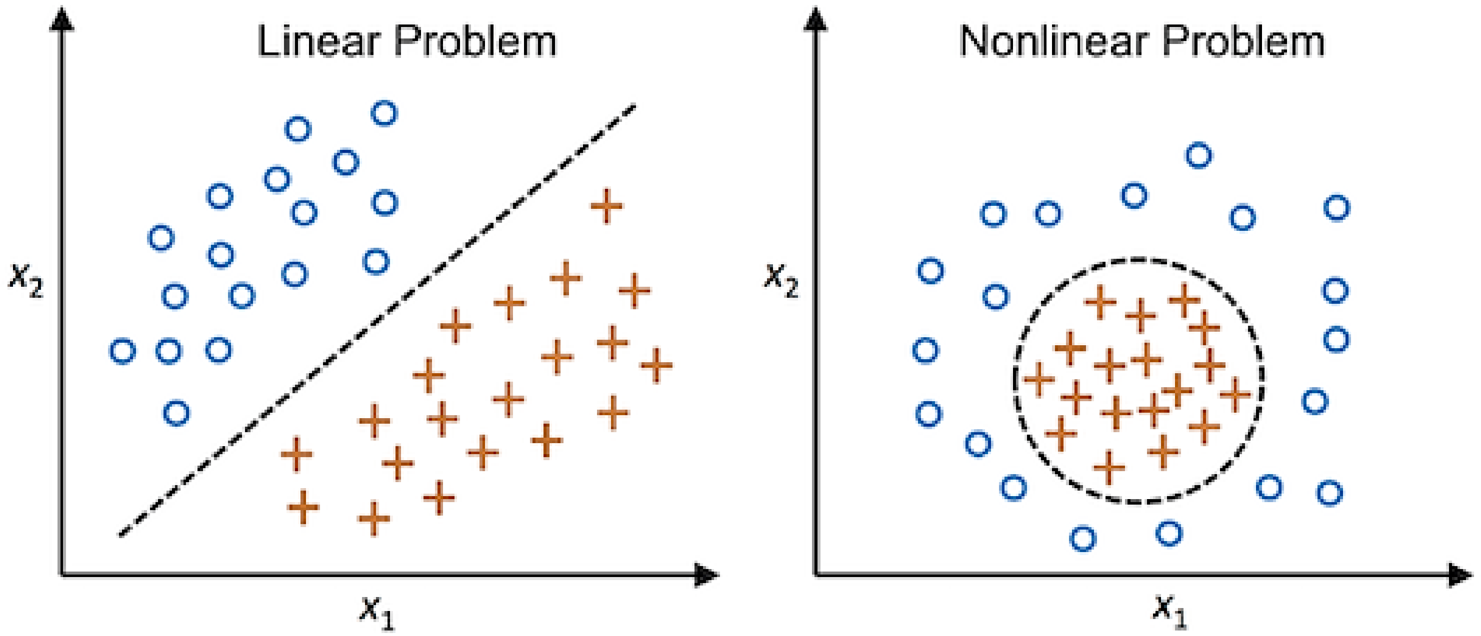
$$Y = \boxed{f}\left(\sum_i \boxed{W_i}\boxed{X_i} + b_i\right)$$

Activation function

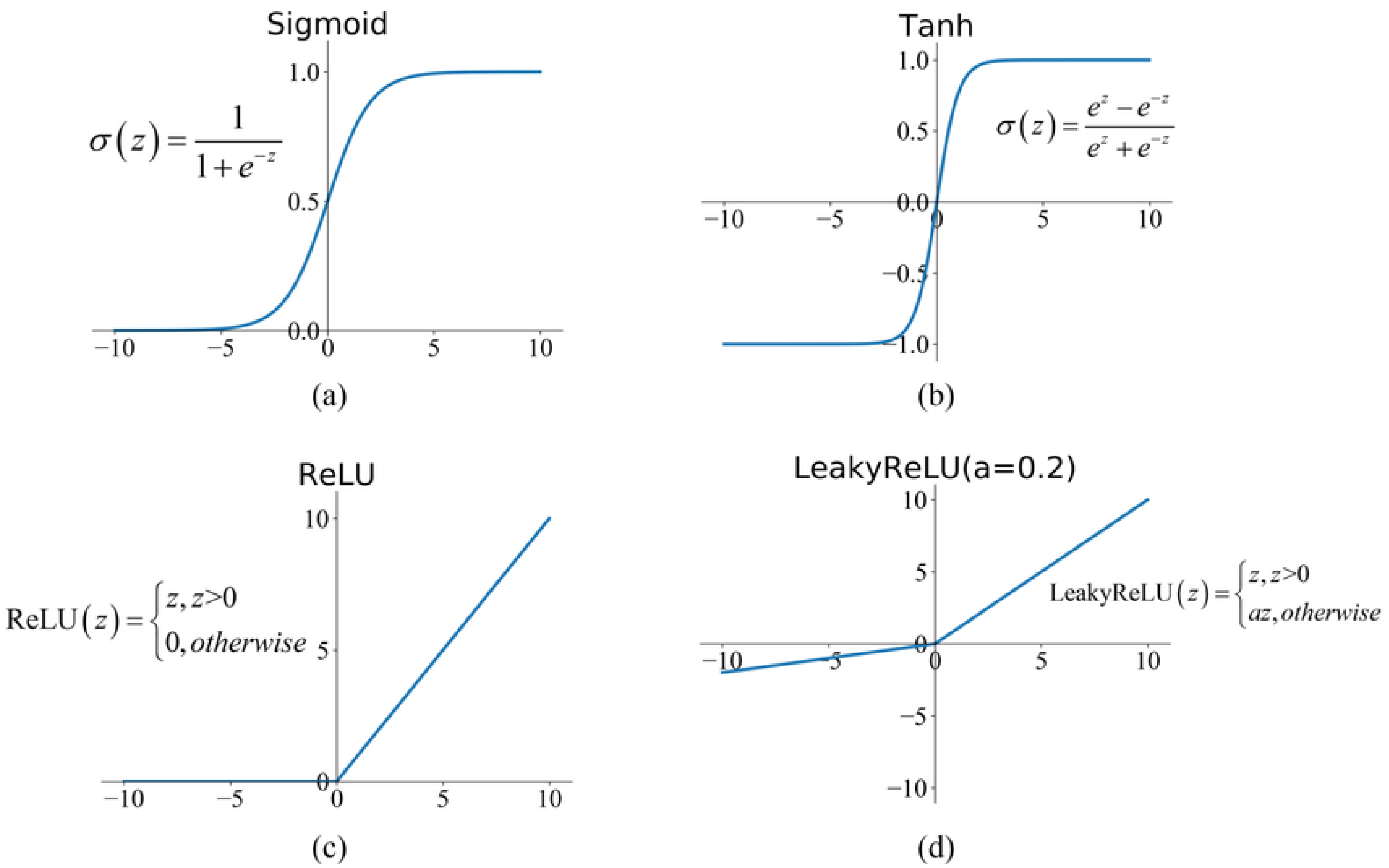
Why Activation functions?

Why Activation functions?

- They introduce **non-linearity** into the network!



Activation functions

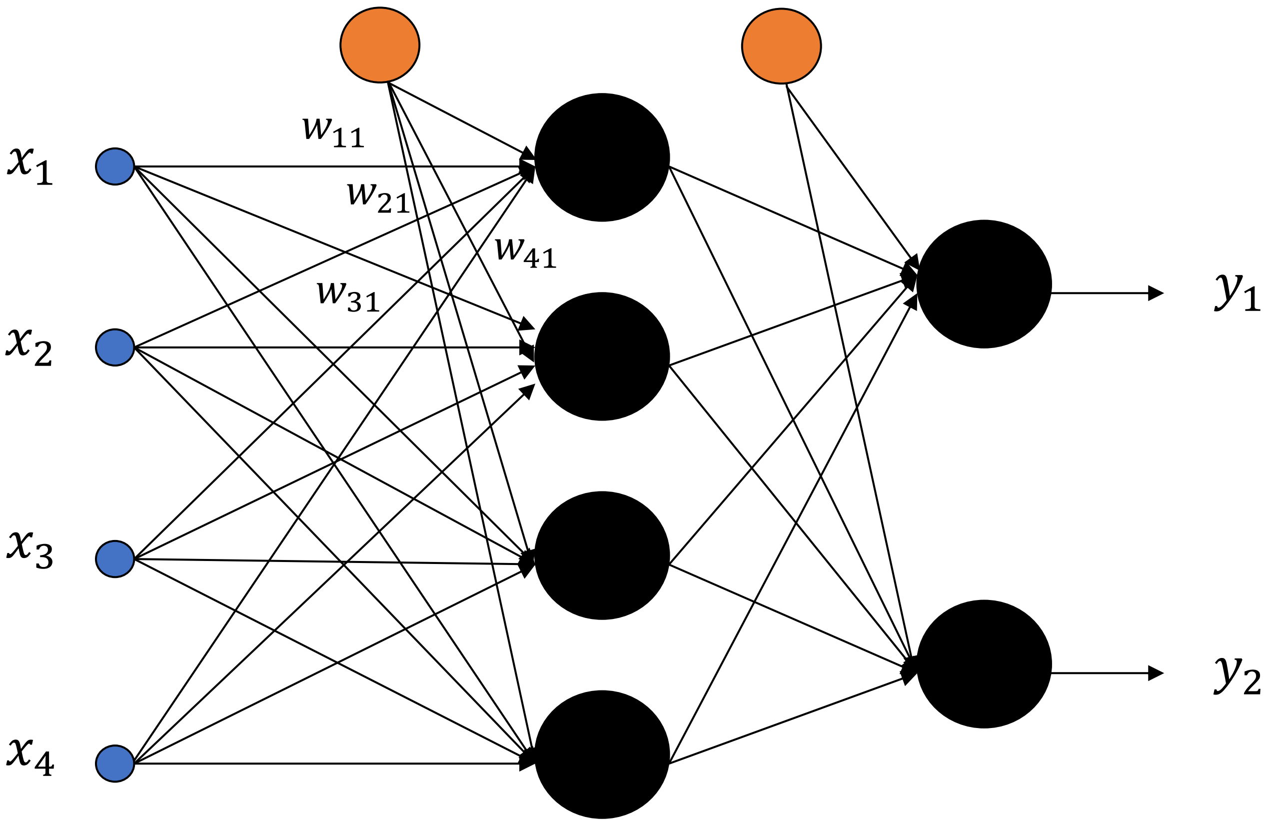


Which one to use?

- **Relu** & **LeakyRelu** are most used activation functions.
- **Dying Relu problem!**
- *LeakyRelu is better!*

$$\max(ax, x)$$

A Simple Neural Network (NN)



$$W_1 = \begin{bmatrix} w_{11} & \cdots & w_{41} \\ \vdots & \ddots & \vdots \\ w_{14} & \cdots & w_{44} \end{bmatrix} \qquad W_2 \rightarrow (4, 2)$$

$$Y = f\left(\sum_i W_i X_i + b_i\right)$$

$$\underbrace{L(\mathbf{w}, \mathbf{b})}_{\text{Error = Loss value}} = \frac{1}{2} \sum_i \big(\underbrace{\mathbf{Y}(\mathbf{X}, \mathbf{w}, \mathbf{b})}_{\text{Predicted label}} - \underbrace{\mathbf{Y}'}_{\text{True label}}\big)^2$$

Backpropagation

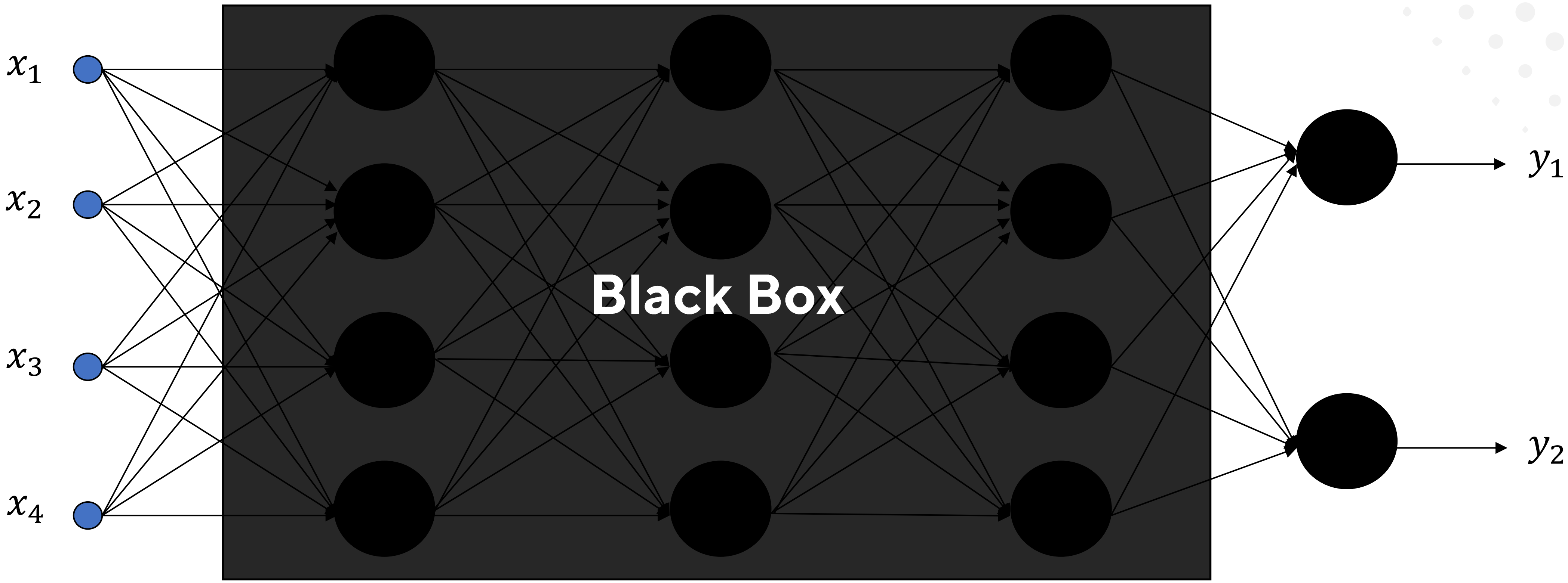
Using optimizers

NN Optimizers

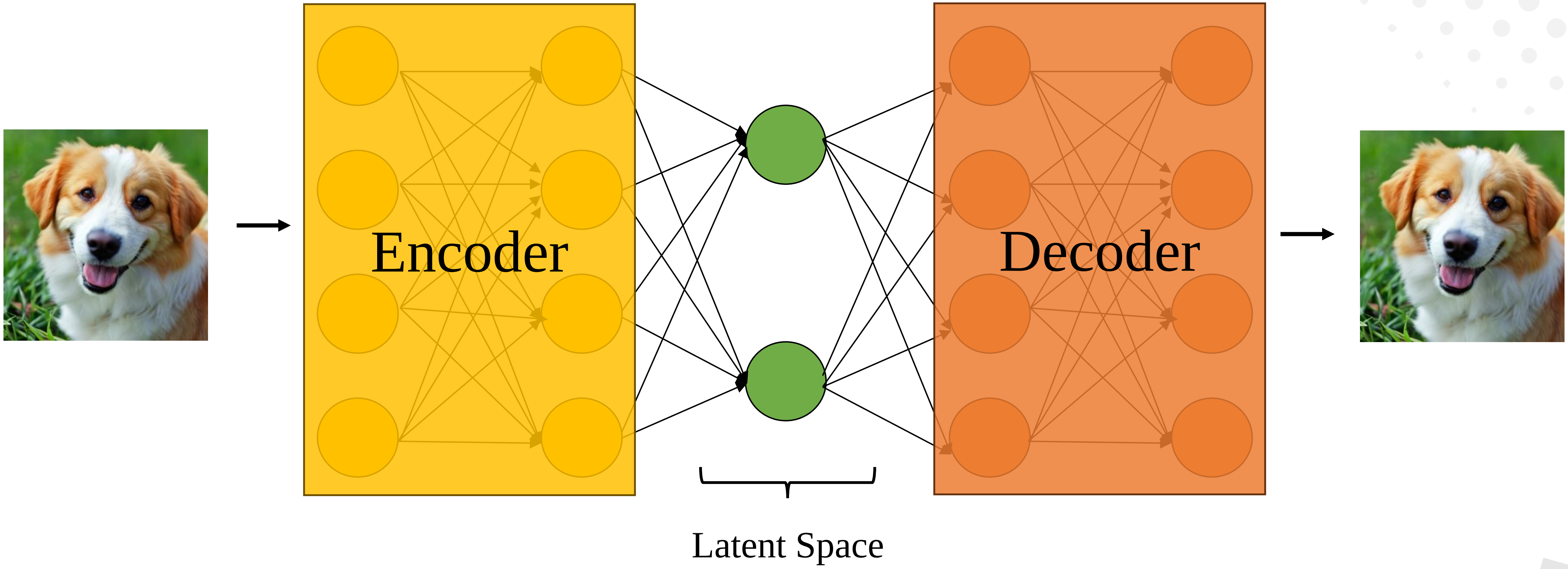
- Optimizers adjust weights of NN to minimize the loss during training
  - ✓**Adam** (Adaptive Moment Estimation)
  - ✓**SGD** (Stochastic Gradient Descent)

→ Selection depends on **model complexity, dataset size, and convergence behavior**.

Deep Neural Network (DNN)

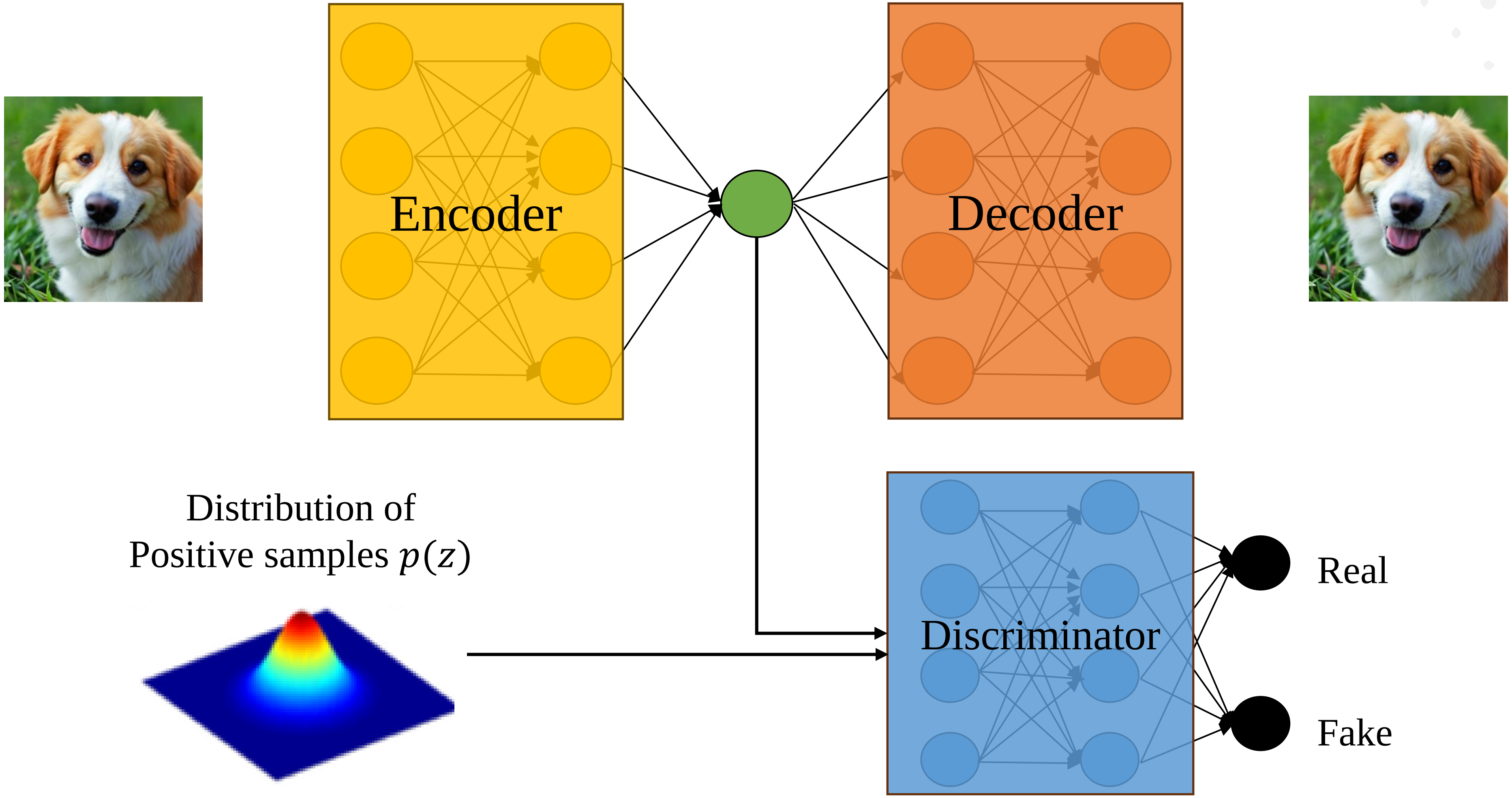


Autoencoders (AE)

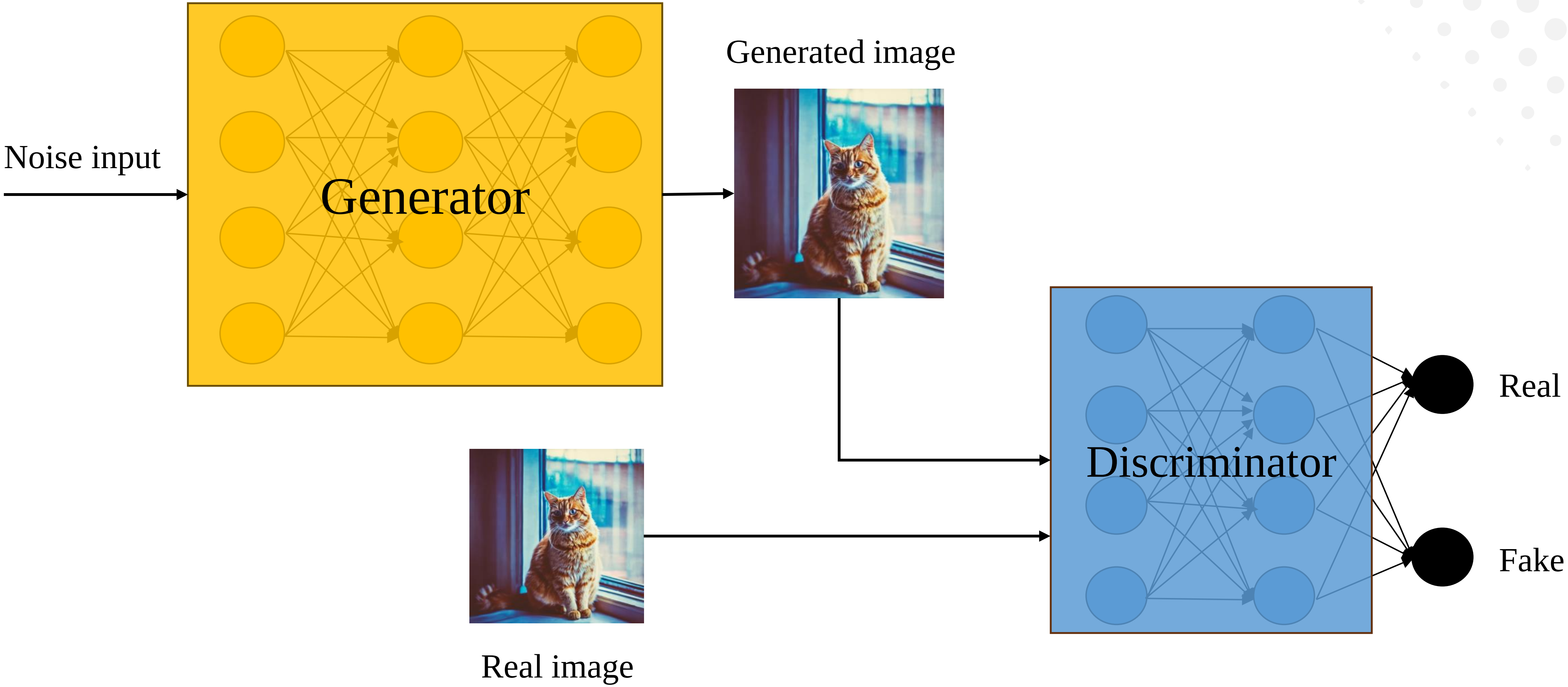


Shiraz University

Adversarial Autoencoders (AAE)

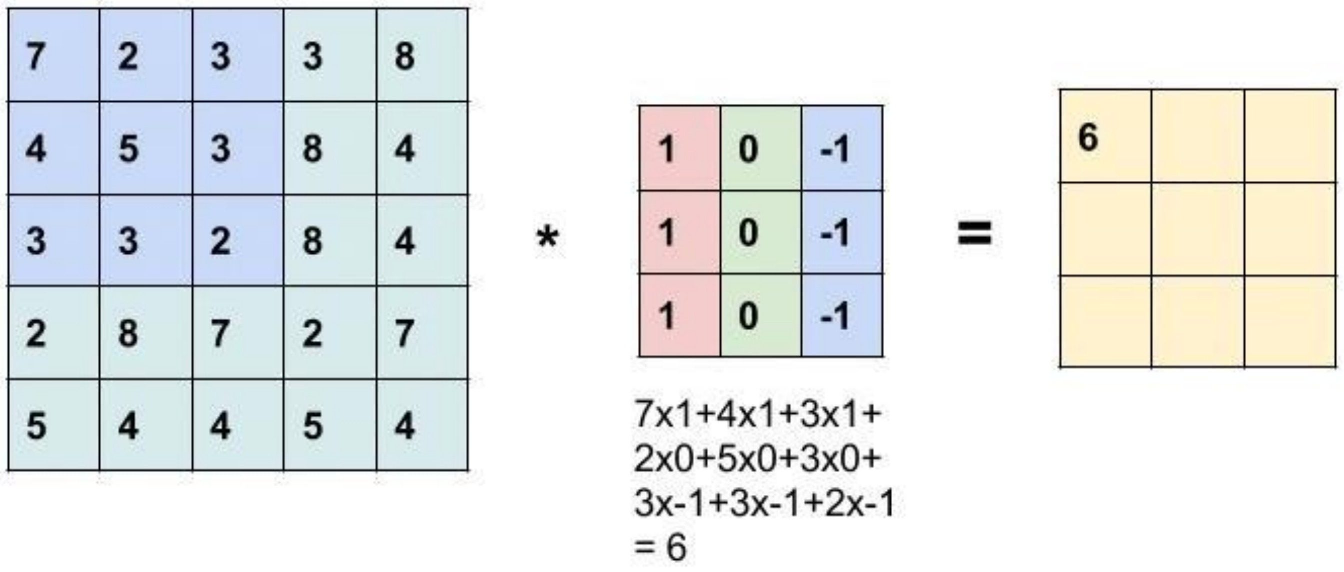


Generative adversarial networks (GANs)



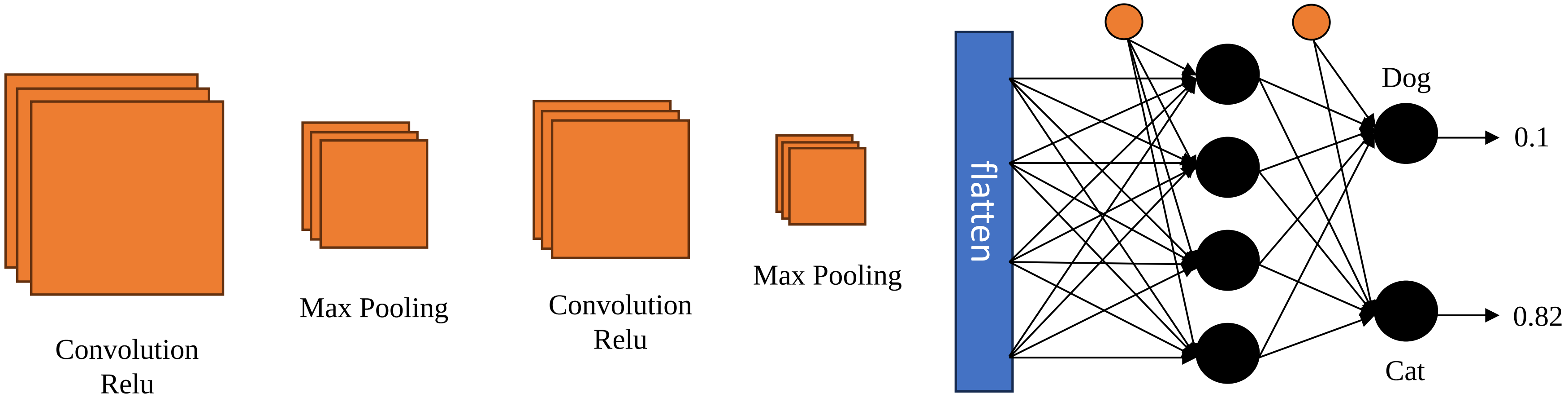
Convolution in Image Processing

- **Convolution:** Transforming an image by applying a kernel over each pixel and its local neighbors across the entire image.
- **Live examples**: https://setosa.io/ev/image-kernels/



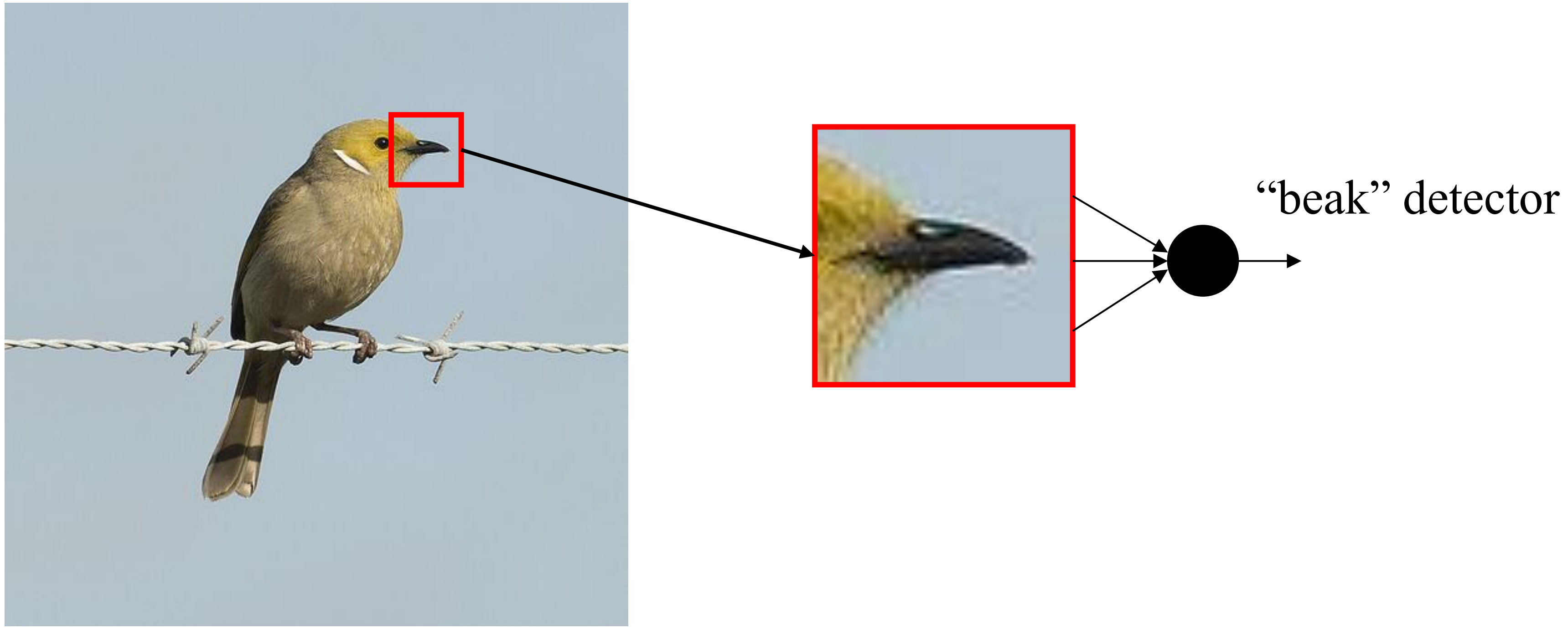
Convolutional Neural Networks (CNNs)

- **Convolution filters** as **<u>feature extractors!</u>**
- https://poloclub.github.io/cnn-explainer/



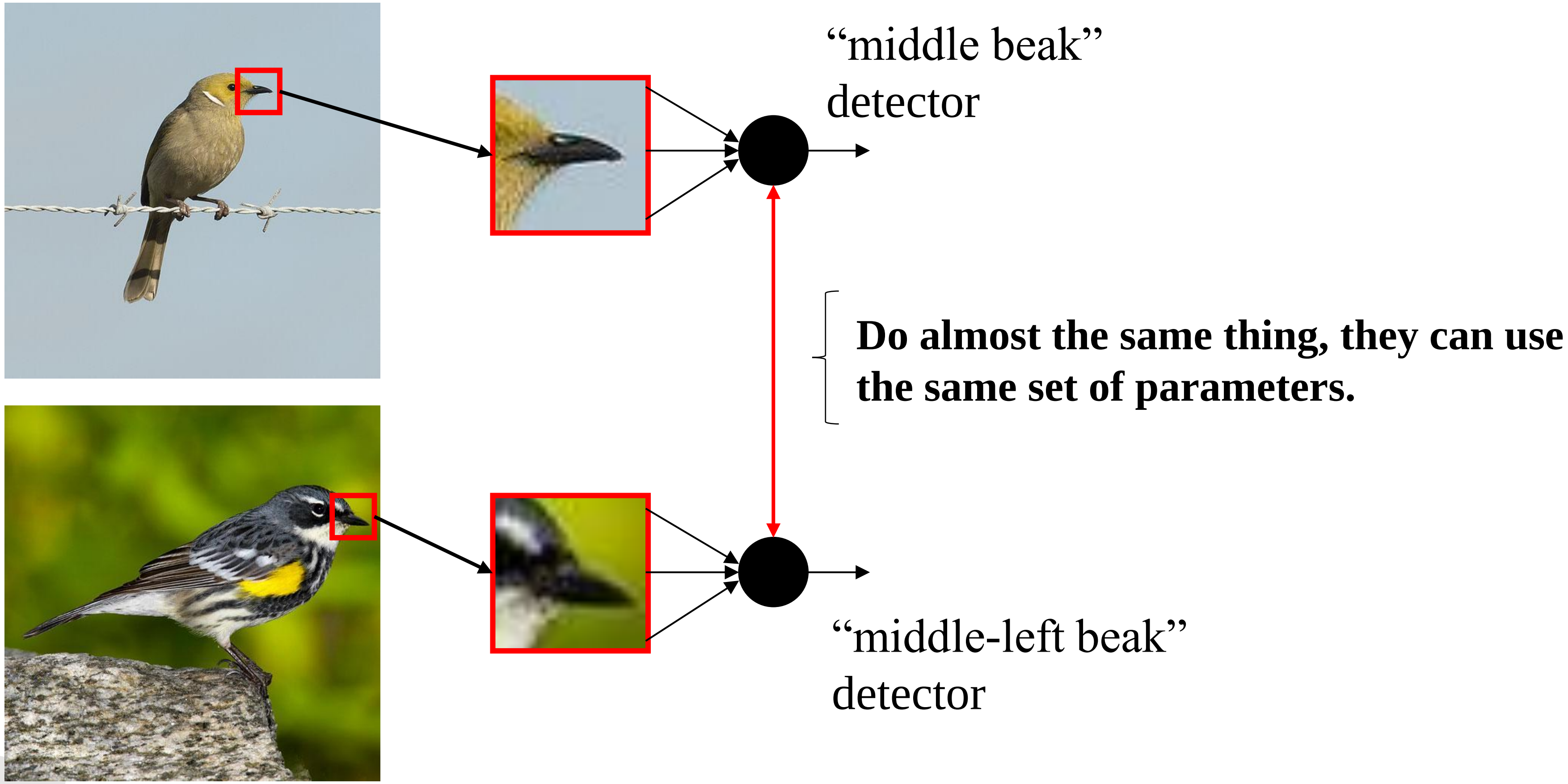
Why CNN for Image?

- Some patterns are much smaller than the whole image
  - ✓ **A neuron does not have to see the whole image to discover the pattern.**



Why CNN for Image?

- The same patterns appear in different regions.



Why CNN for Image?

- **Subsampling** the pixels will not change the object

bird

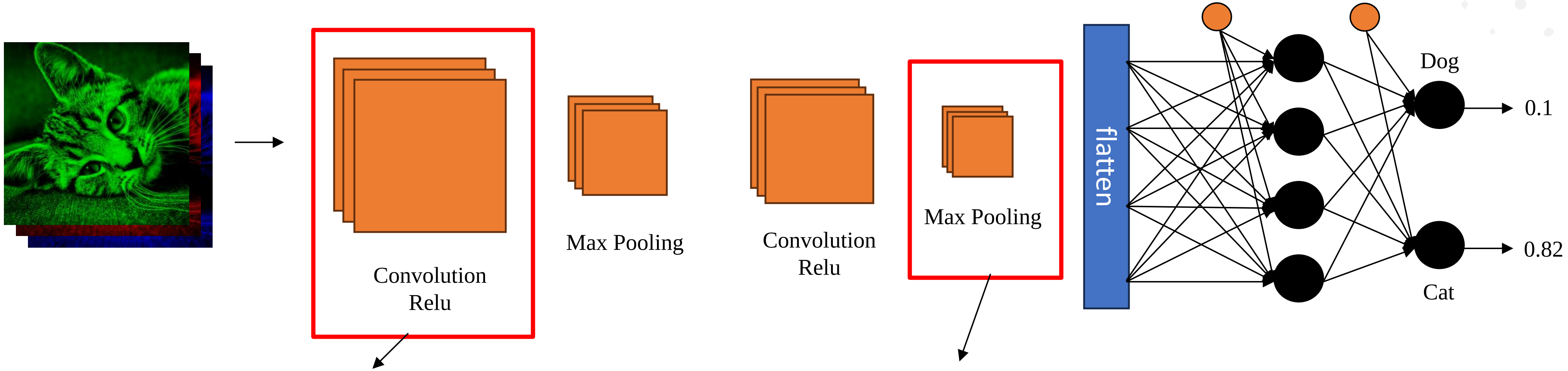
Subsampling

bird

- We can subsample the pixels to make image smaller
- Less parameters for the network to process the image

Convolutional Neural Networks (CNNs)

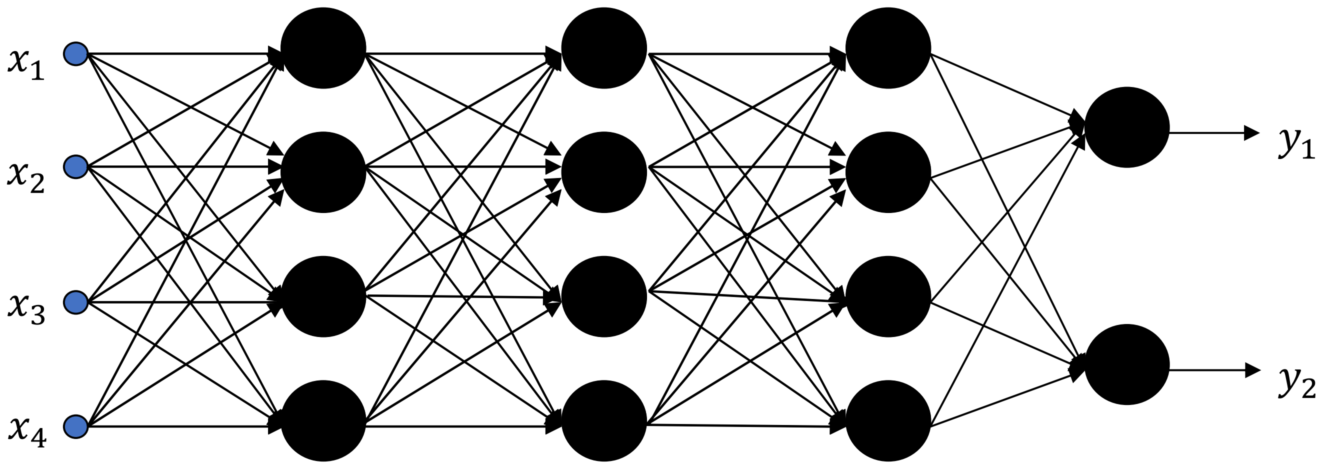
- **Convolution filters** as **<u>feature extractors!</u>**



- Some patterns are much smaller than the whole image
- The same patterns appear in different regions.

- Subsampling the pixels will not change the object

Transfer Learning



Training a huge model on a large dataset for a very long time!